

# Mindtree

*Welcome to possible*



## Virtual Desktop Infrastructure (VDI) on Premise Storage and Beyond SNIA -Storage Developer Conference India May 2017

Agile | Digital | Enterprise Applications | Managed Services

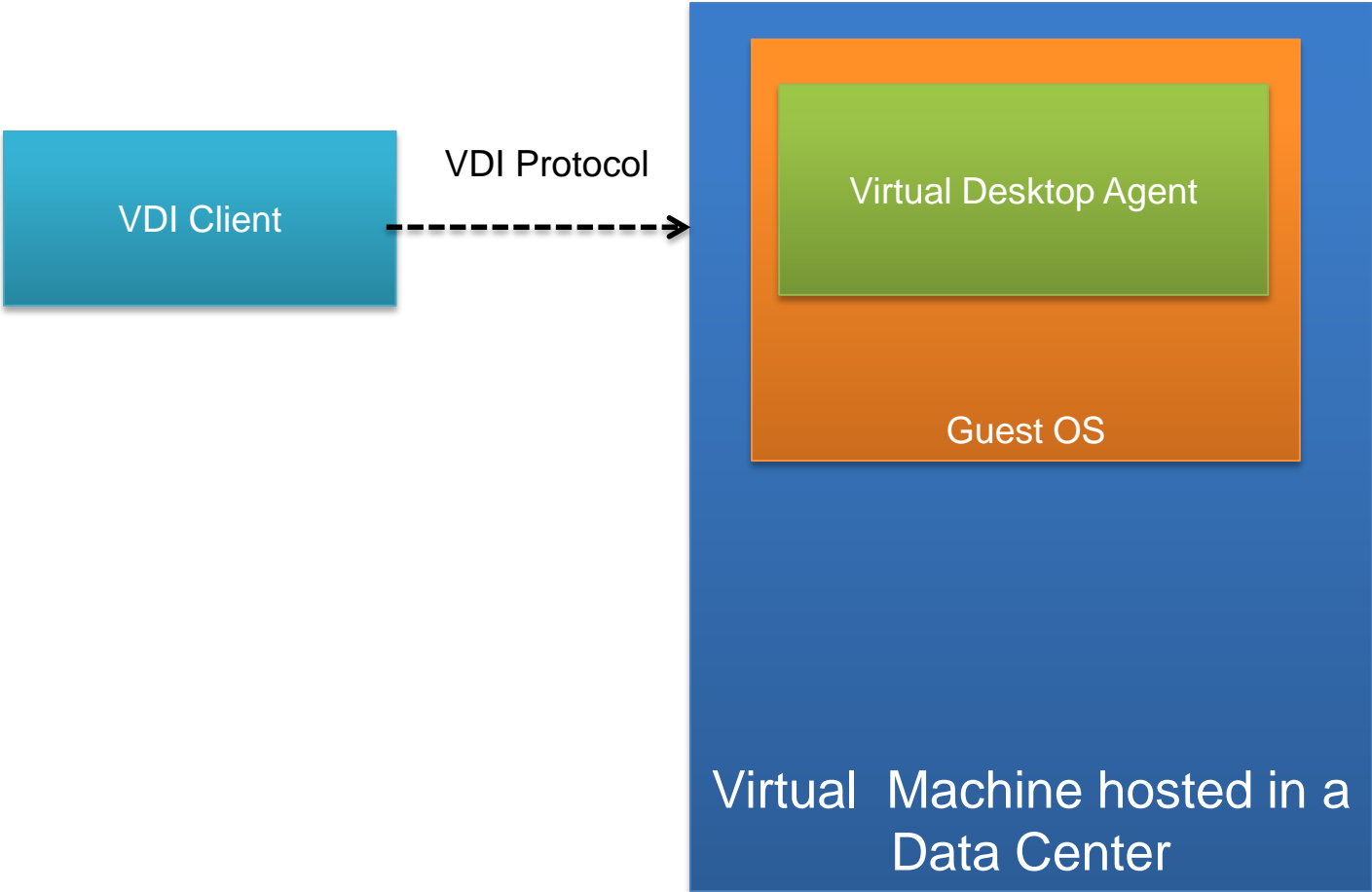
# Agenda

- What is VDI and why it is required
- Aspects related to VDI performance
- VDI on HCI
- Future trends
- Conclusion

# What is Virtual Desktop Infrastructure (VDI)?

Virtual desktop infrastructure (VDI) is virtualization technology that hosts a desktop operating system on a centralized server in a data centre. VDI is a variation on the client-server computing model, sometimes referred to as server-based computing.

# VDI Components

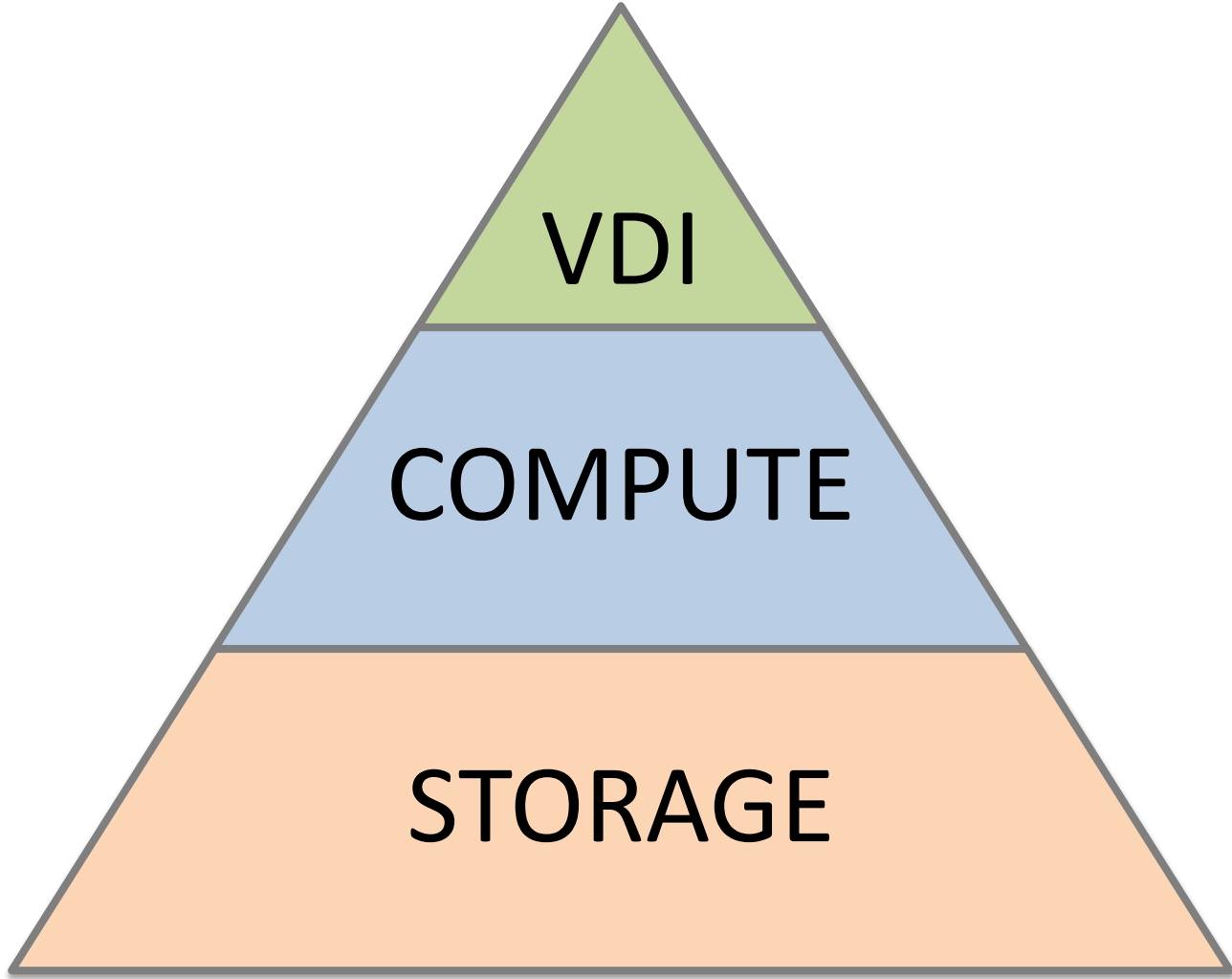


# VDI Components

How are Desktops Delivered

Where Desktops Run

Where Desktops Live



# Why VDI?

## Improved Security and easier Management

- More secure as data resides in the Data Centre
- Any device can be used to access VDI – PC, Thin Clients, Tablets, Mobiles
- Easier to manage - there is no need to lock down the end-point device
- Centralised backups and DR

## Lower Cost

- Non-persistent virtual desktops can be used for knowledge workers, needing lesser resources
- Lower licensing cost due to concurrent licensing mode
- Lower power and space cost
- Lesser staff needed to manage the infra including desk side assistance

# Why VDI? (contd.)

## Deployment

- Faster deployment and accelerated time to value
- Hundreds of new desktops can be provisioned in a few minutes

## Performance

- With modern Graphics, Storage and Networking hardware, VDI can perform as good or even better as compared to a physical desktop
- VDI configuration can be scaled up or down as needed

## Additional benefits of VDI

- Green Computing Initiative
- Needs of a mobilised workforce
- HA/DR for Desktop estate
- Hardware Independence



# Scenarios where VDI is useful

## High Security and regulated environments

Verticals such as BFSI, Pharma and Military may have the need to maintain absolute control over user desktops

## Dev and Test

- VDI is ideal for test and Dev environments, where there is a need to frequently rebuild desktops to the standard build

## Distributed users

- Ideal for organization whose user base is widely distributed
- VDI is centrally managed and the end users can use a thin client or a personal device to access VDI

## Few concurrent users for expensive software

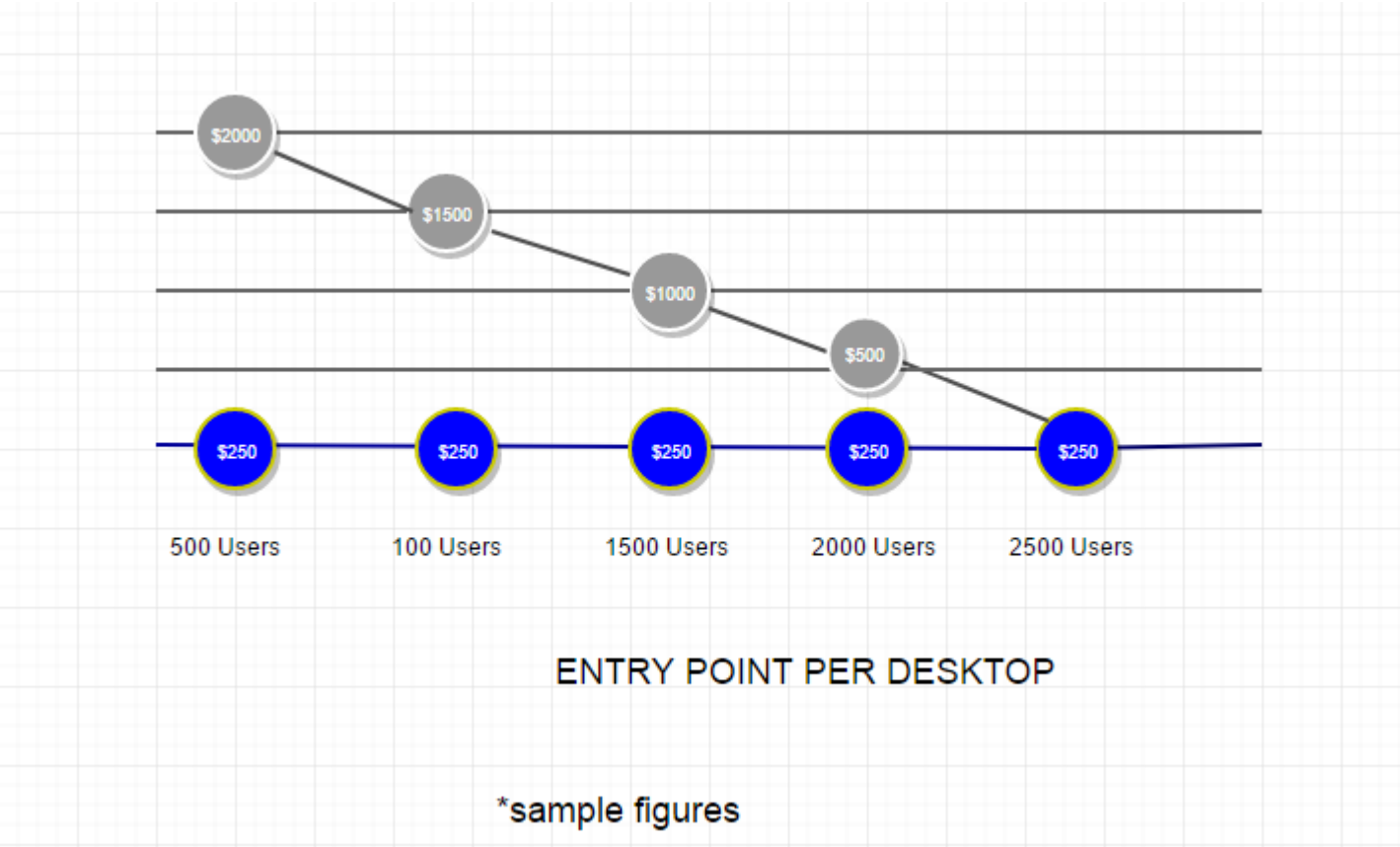
- Number of concurrent users is considerably lesser than the number of licensed users

## Why VDI projects failed in the past?

- Performance was not as good as physical desktops
- SAN storage is too expensive in order to get sufficient IOPS
- Underestimating Network requirements
- Underestimating the Complexity - VDI is not as simple to implement and manage as initially estimated
- Failure to identify candidates for App Virtualization v/s VDI
- Some legacy applications cannot be virtualised, rebuilding apps for VDI is too expensive
- VDI licensing is not properly understood/estimated
- Ultimately, for the same performance as physical desktops, VDI is not much cheaper

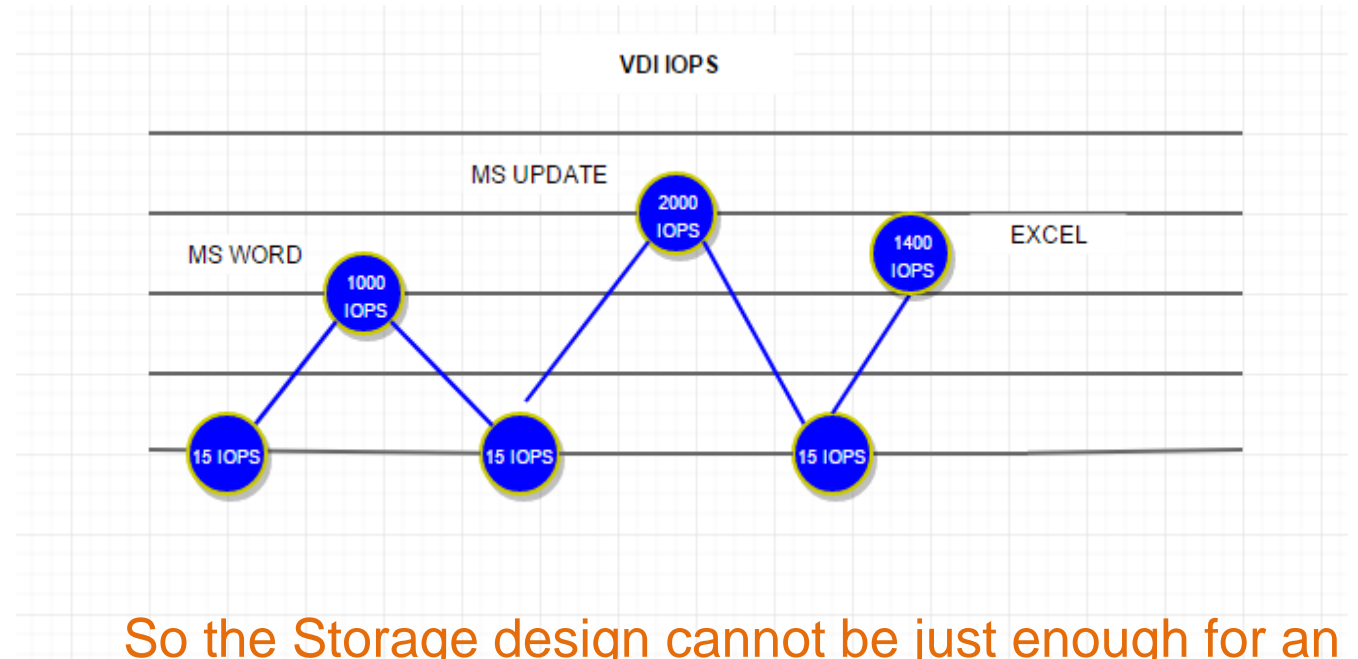
# Considerations for VDI

- Entry point
- Scalability
- Performance
- Monitoring
- Capacity



# VDI IOPS

- A virtual desktop workload is very different from other types in DC
- VDI is very spiky in nature
- Opening an application the first time in a session can generate huge IOPS
- Way beyond the average IOPS
- Unlike many server desktop workloads, VDIs are typically write heavy

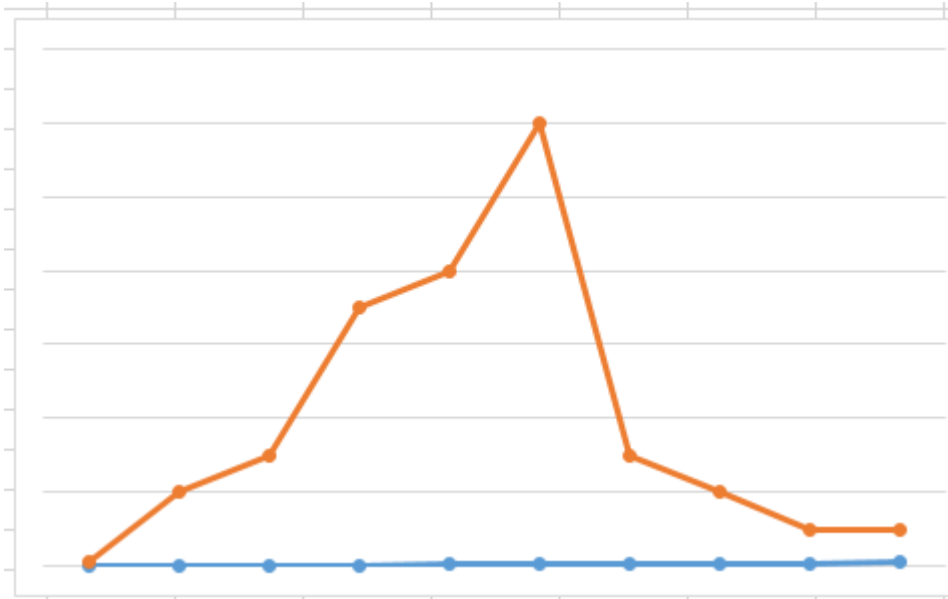


So the Storage design cannot be just enough for an average I/O. It should consider peaks (including booting and login events)

# Boot Storms and Login Storms

Two of the taxing events on the storage tier are the boot storm and the login storm

- Booting a virtual desktop requires that the key OS bits be loaded from the SAN
- Login storm can be usually 30-50% less IOPS than boot storm
- Timed Boot-Boot storm can be controlled by starting machines during off-peak hours



# Baseline – I/O Profiles

User Types	Task Performed	IOPS (per concurrent user)
Light user	Working in a single application and is not browsing the web	~6
Normal user	Working in a few applications with minimal web browsing	~10
Power user	Runs multiple applications concurrently and spends considerable time browsing the web	~25
Heavy user	Tasks that have high I/O requirements like compiling code or working with images or video	~50

# Calculating Workload IOPS

- Login IOPS = MaxSimultaneousUsers \* Average Login IOPS
- Workload IOPS = MaxSimultaneousUsers \* Average Workload IOPS
- Peak IOPS = Workload IOPS + Login IOPS
- Boot Storm Size = Total IOPS available/300
- Login Storm Size = Total IOPS available/100

Most of the environments are mix of user types mentioned previously

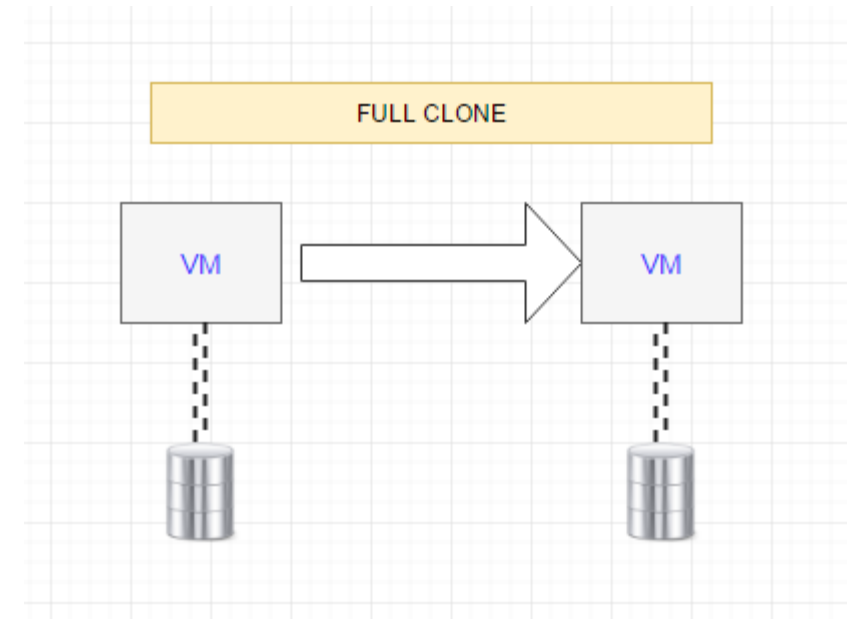
- Taking those proportions, a “loaded” rate for the environment can be calculated
- Loading IOPS = Light (a\*6) + Normal (b\*10) + Power (c\*25) + Heavy (d\*50) = Total

\*If you have the ability to perform a pilot, you can gather data and analyze the user’s actual IOPS

# Full Clone

There are a mainly two ways of provisioning desktop

Full clones - A full clone is an independent copy of a virtual machine that shares nothing with the parent virtual machine .

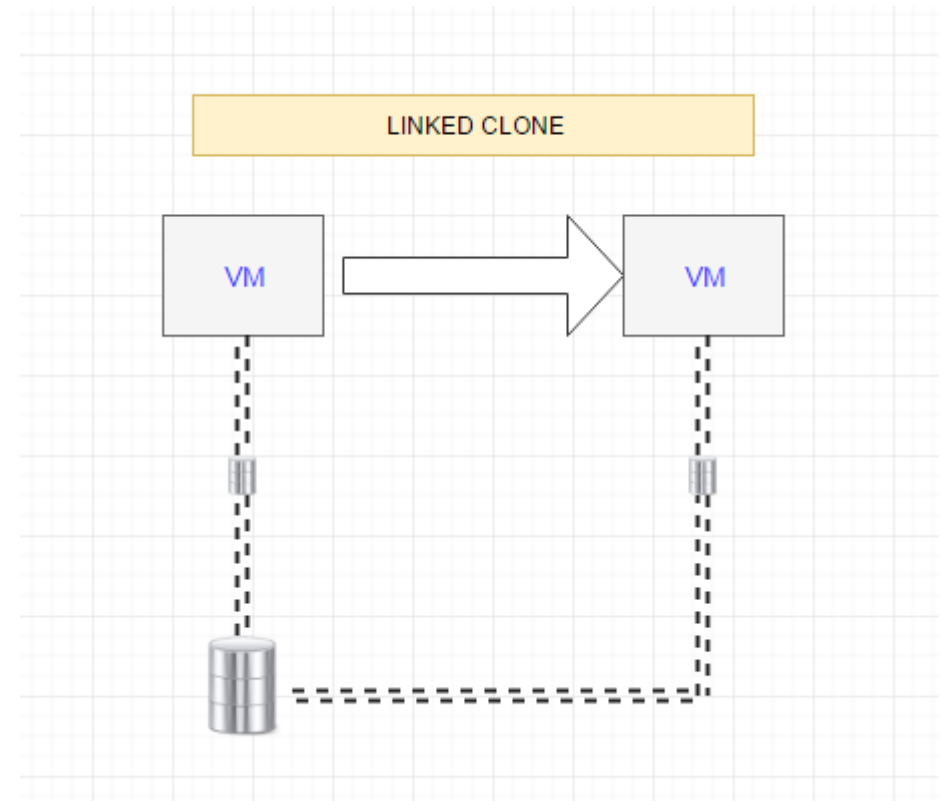




# Shared Image

Shared image approaches require less storage capacity

- Parent image is shared and each virtual desktop is only consuming a smaller amount of space for its unique data.
- This image is now used by hundreds or thousands of virtual desktops



# Full Clone vs Shared Images

- More Storage space required
- Deduplication will be important.
- Full clones must also be patched independently, which will increase the I/O during those operations.
- Must be able to generate large amounts of IOPS to handle boot storms.
- If the image is a bottleneck, all virtual desktops using it will be negatively affected

# Persistent vs Non Persistent Virtual Desktops

- A non-persistent virtual desktop does not retain any data on the desktop itself after a logoff or reboot.
  - This includes any data such as user settings, application settings etc.
  - User settings may be stored in a central location and applied to any desktop they logon to.
- 
- A persistent desktop retains data on the desktop itself between logons and reboots
  - This includes all data such as user settings, applications etc.
  - User can have an application installed directly on their desktop

# Storage Choices

- All-Flash
- Hybrid Flash
- Hyperconverged Infrastructure

# All Flash Storage

- Entirely made up of flash-based storage
- Modern AFAs were designed with flash in mind
- Legacy array simply replaced the spinning disks with all-flash
- Very fast, with only one level of performance in the product
- To get the capacity look for effective deduplication and compression



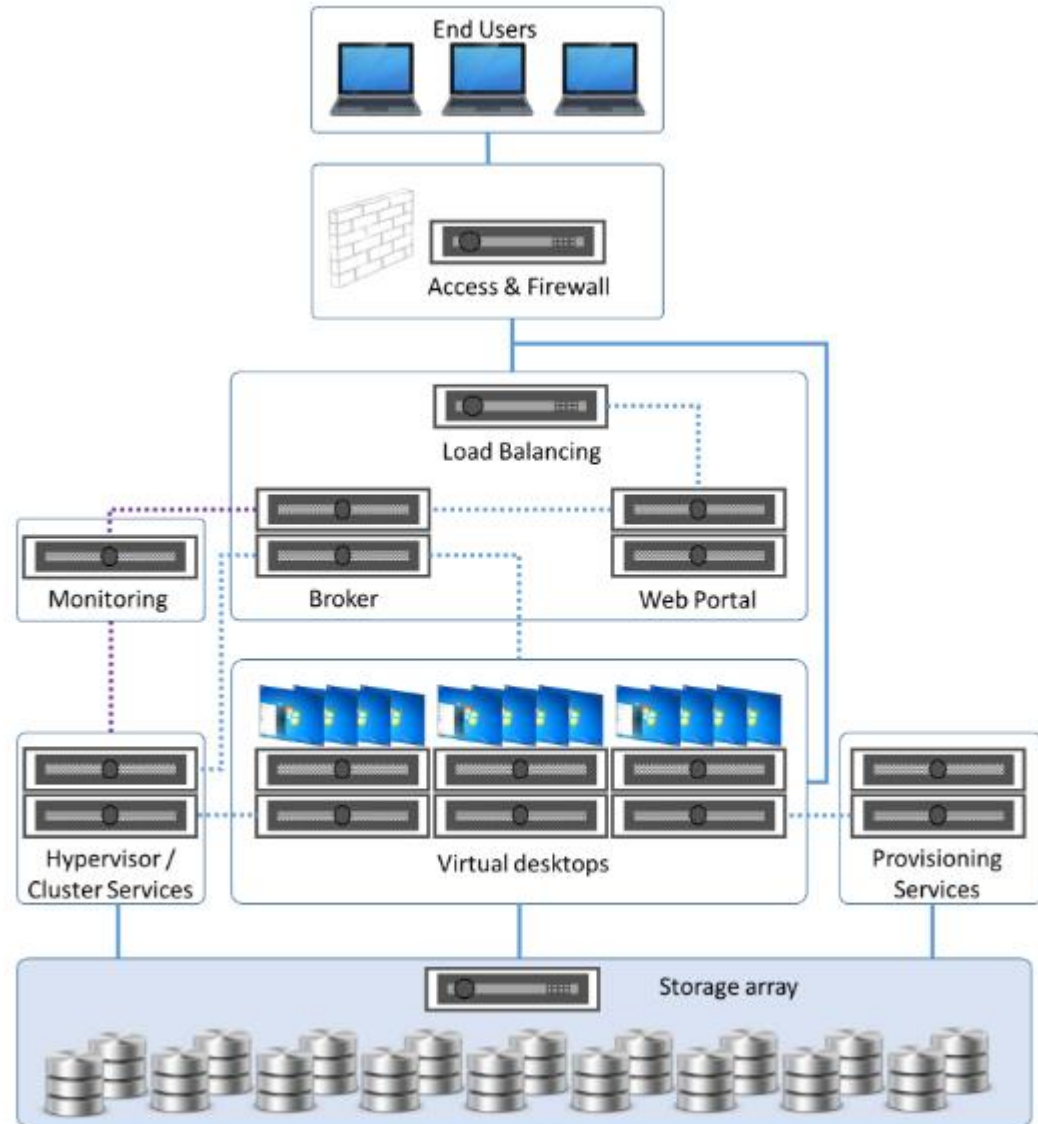
# Hybrid Storage

- Modern architectures to efficiently use a combination of flash drives and spinning disks.
- Impressive performance from a smaller amount of flash
- Large amount of capacity by storing data on large spinning disks in the array
- Intelligence to automatically tier data across flash and disk drives based on demand

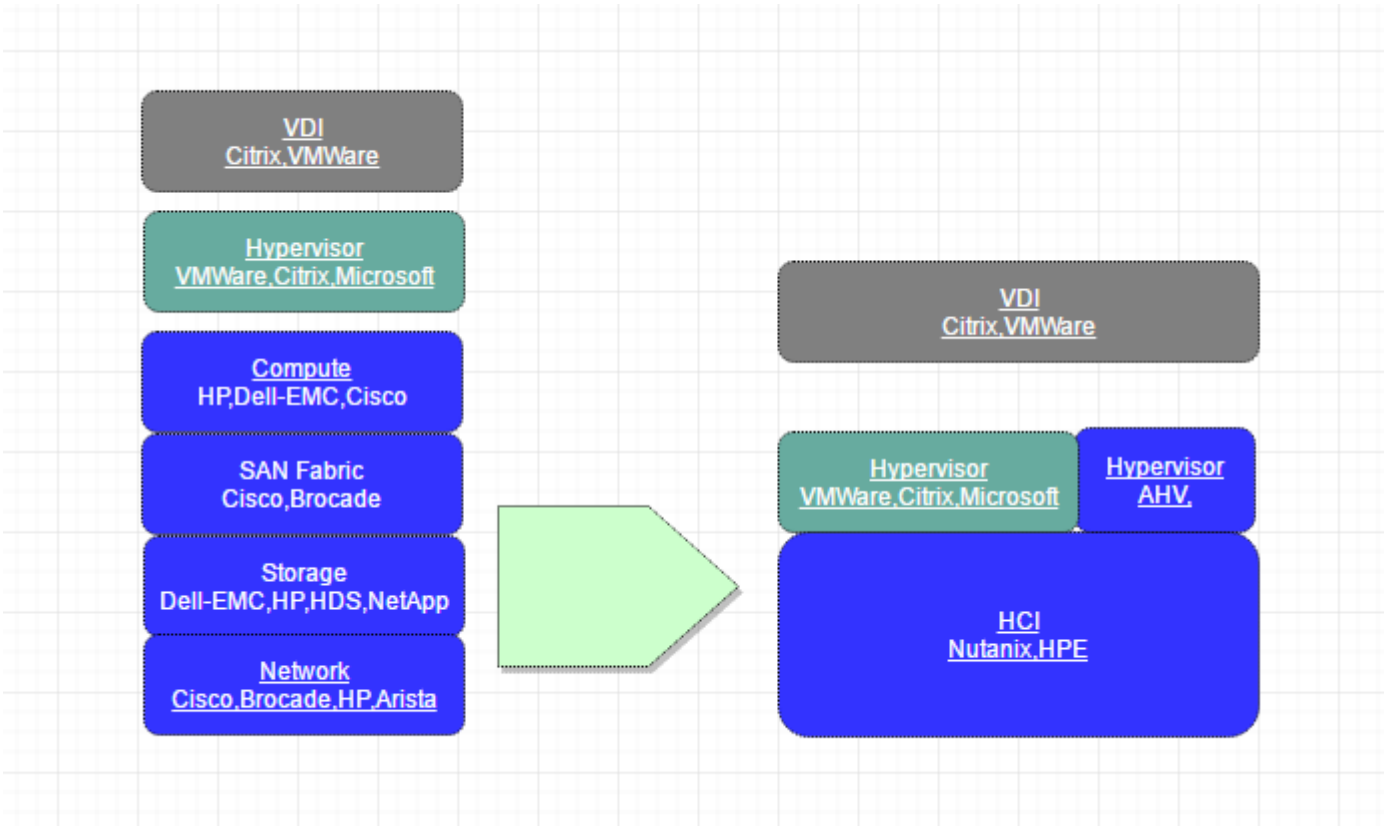


# Traditional VDI Architecture

- Firewall and Load Balancers
- Web Portal
- Connection Broker
- Monitoring
- Virtual Desktops
- Hypervisor
- Provisioning services
- Compute and Storage
- Storage networks connects compute and Storage



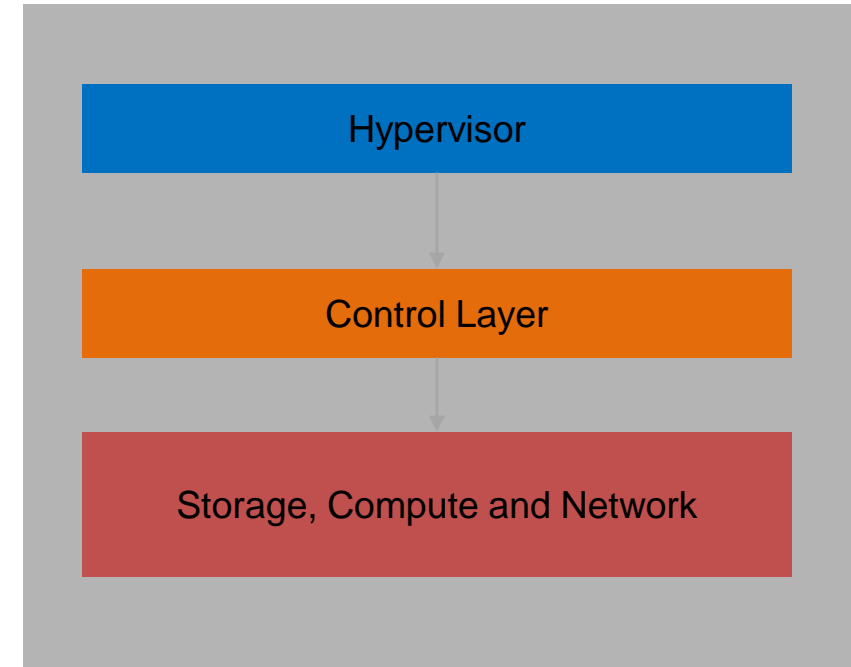
# VDI on HCI





# Hyper-Converged Infrastructure (HCI)

- Storage, Compute, Network and virtualization together integrated managed by a single software.
- Easily scalable -“Lego” blocks
- Leverages commodity hardware
- Simplified Management
- Software-centric approach
- Can be used to provide various cloud services like IaaS, PaaS, SaaS
- Provides features like Inline Compression, Data Deduplication and Data Optimization



Consumption model for software-defined storage

# HCI and converged infrastructure (CI)

## Similarities

- Both are deployed as blocks
- Both converge critical resources to deliver higher levels of density

## Difference lies in

- HCI has direct integration with the virtualization architecture
- Software-defined infrastructure built on top of physical components

# Benefits of HCI

A truly Hyperconverged product offers a number of benefits

- Simple Installation – Automated process
- Easy Scalability – easy to scale up or down
- Modern management – VM as point of management
- Extensibility – Integrate the infrastructure with other parts of the solution. (API-based integration)
- Performance

HCI is about creating an infrastructure layer that is simple and efficient

## Why VDI on HCI makes sense?

- Increased utilization of VDI often creates pressure on traditional infrastructure
- Traditional infrastructure comprising discrete components must configure and tune each of those components for the individual workload

And HCI can

- HCI can start small and allows scale out
- Add more node with minimum downtime
- Multiple components are bundled together, including hypervisor

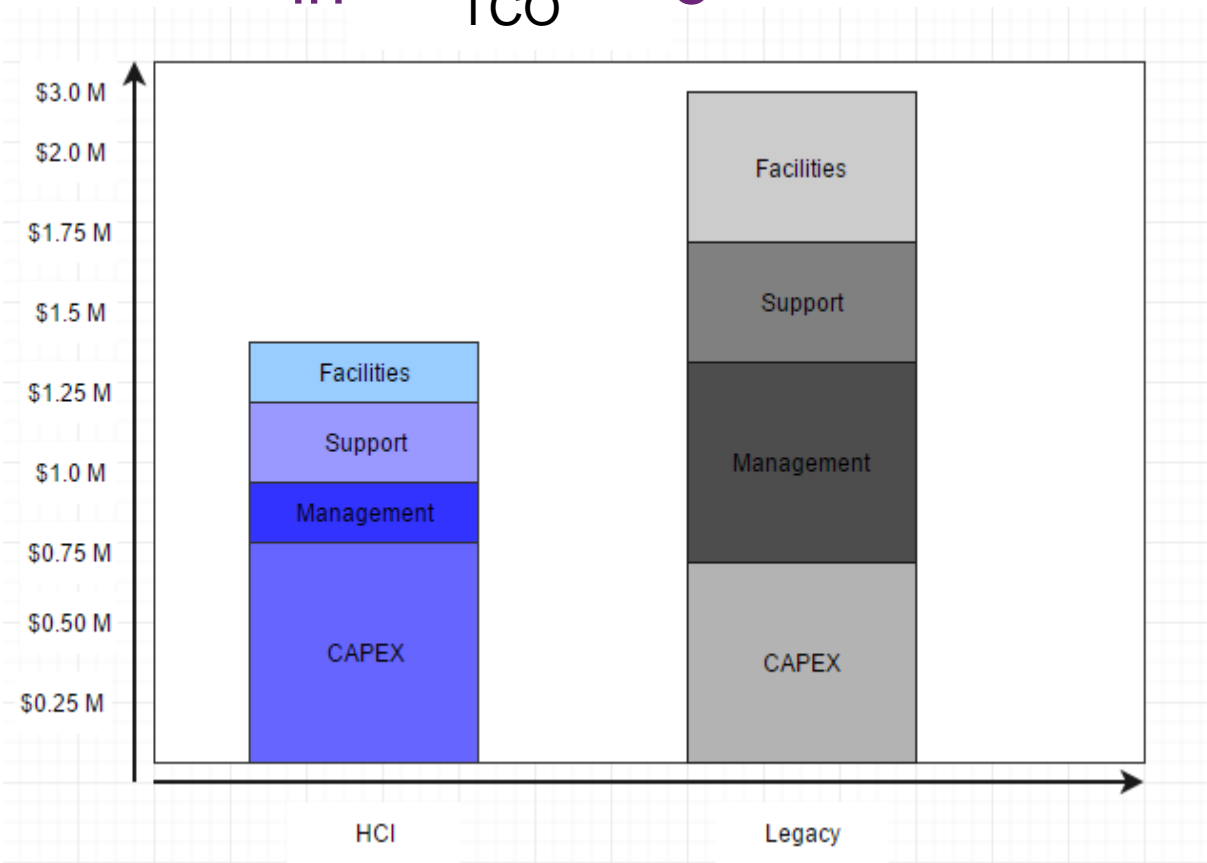
# What to look out for in HCI solution

Not all HC solutions are purpose-built for the highest possible savings in operating costs

- Application-specific reference architectures
- Pre-configured and pre-tested with predictable scale, workloads and cost
- Operating expenses
- Rapid time to value with prescriptive deployment

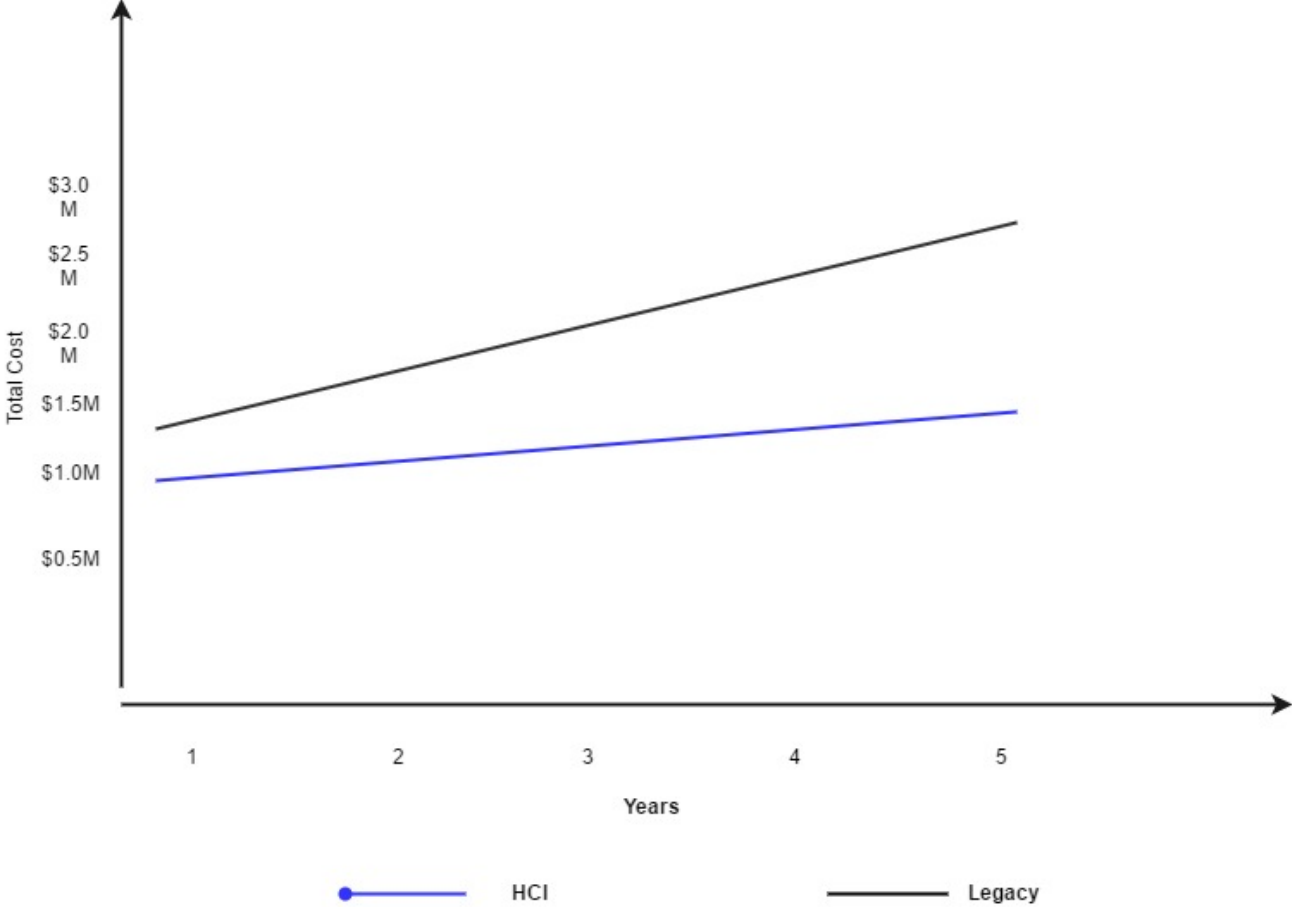
# A case for VDI – TCO – HCI versus legacy 3-tier

in TCO



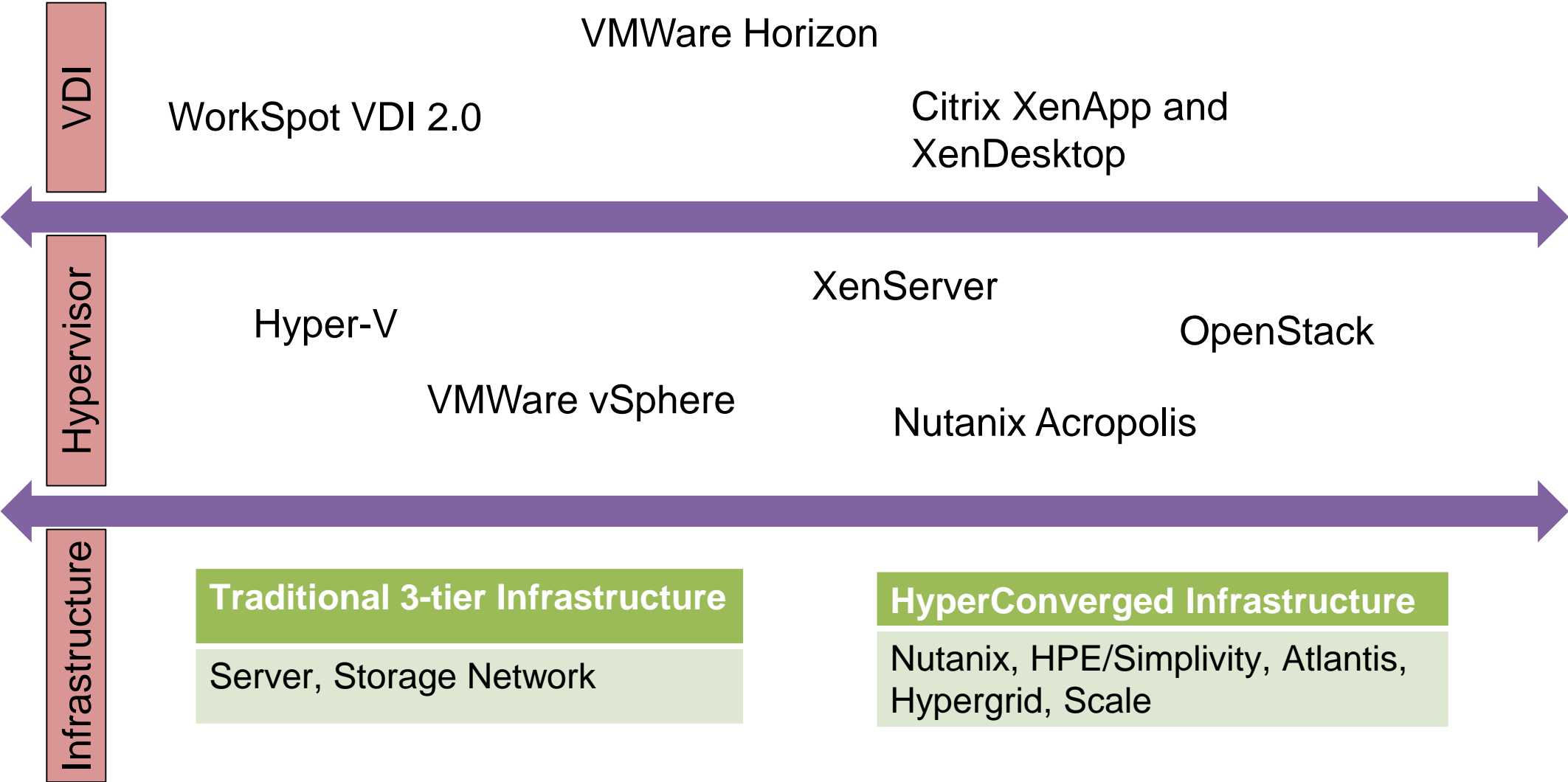
- The illustrated TCO is for a 5 year period

# YoY comparison



- There is approximately a 55% savings in HCI as compared to a traditional 3-tier infrastructure

# VDI – Components (Traditional Infrastructure and HCI)





# What users want

Users want any device to have access to:

- Windows apps
- Network file shares
- SaaS apps (Office 365)
- Web apps (SharePoint)
- Native apps

# What is App Virtualization?

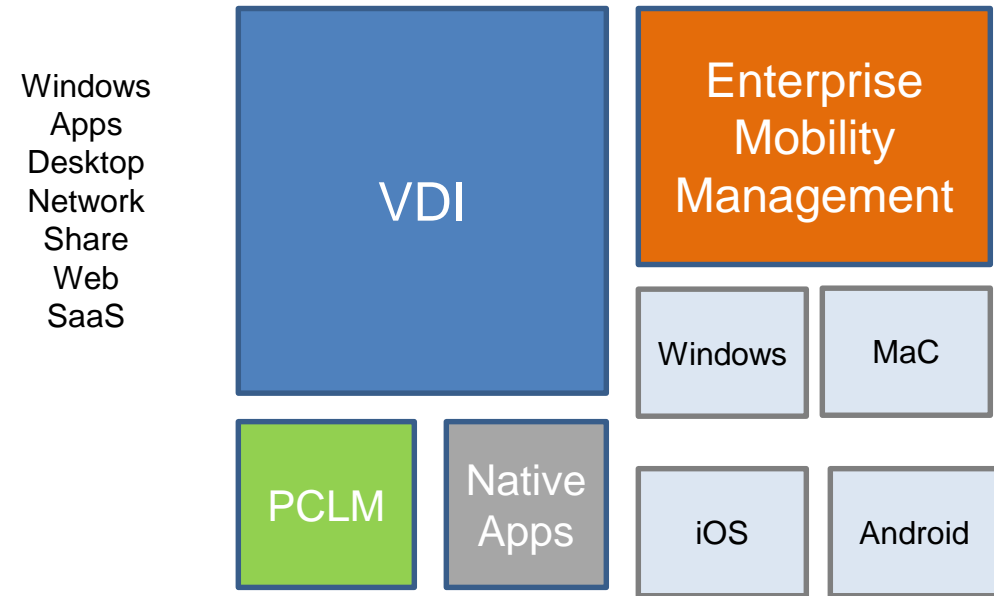
Application virtualization is a technology that enables execution of an Application without installation on a client computer

- **Remote** - Remote applications run on a server. It runs in a separate window on the user's desktop and behaves like a locally installed application
- **Streaming** - Required components of the application are downloaded to the local computer and the application is executed. Additional components may be downloaded as needed.

# Its not just VDI –we want more

Architectures are evolving

- Unified End Point Management
- VDI to be a feature of a workspace
- Seamless productivity across all devices



# Workspace

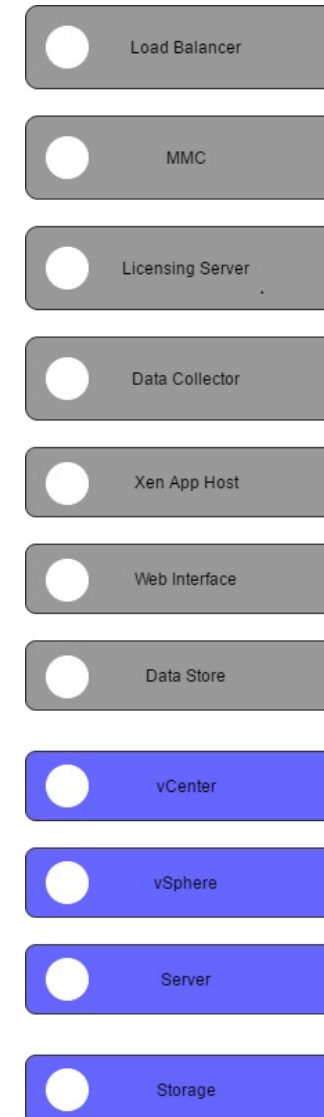
With workspace

- Apps can be in the cloud, or on-premises
- Apps can include Windows/Web/SaaS/Native/HTML5/Hybrid
- Windows desktop can be an “app”

**Secure access to any application from any device**

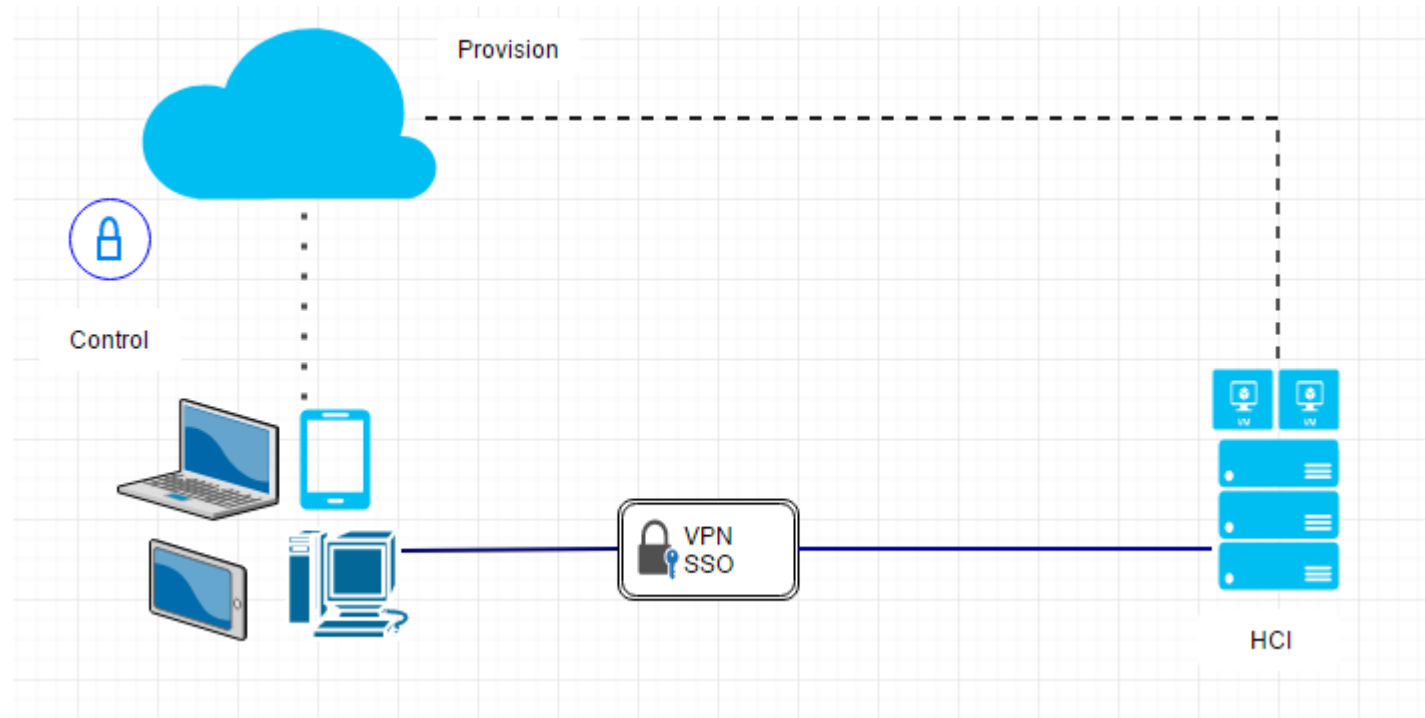
# VDI is complex

- Multiple components to be deployed
- Manage all the components and its high availability
- Increased operational complexity of VDI
- To resolve any issue multiple teams are required – Storage, Virtualization, Server, Network etc.



## New players with VDI solutions on HCI

- Run all the supporting components in the cloud—web portal, connection broker, provisioning, LB
- Actual desktops, apps, users, servers, storage and data live on premises

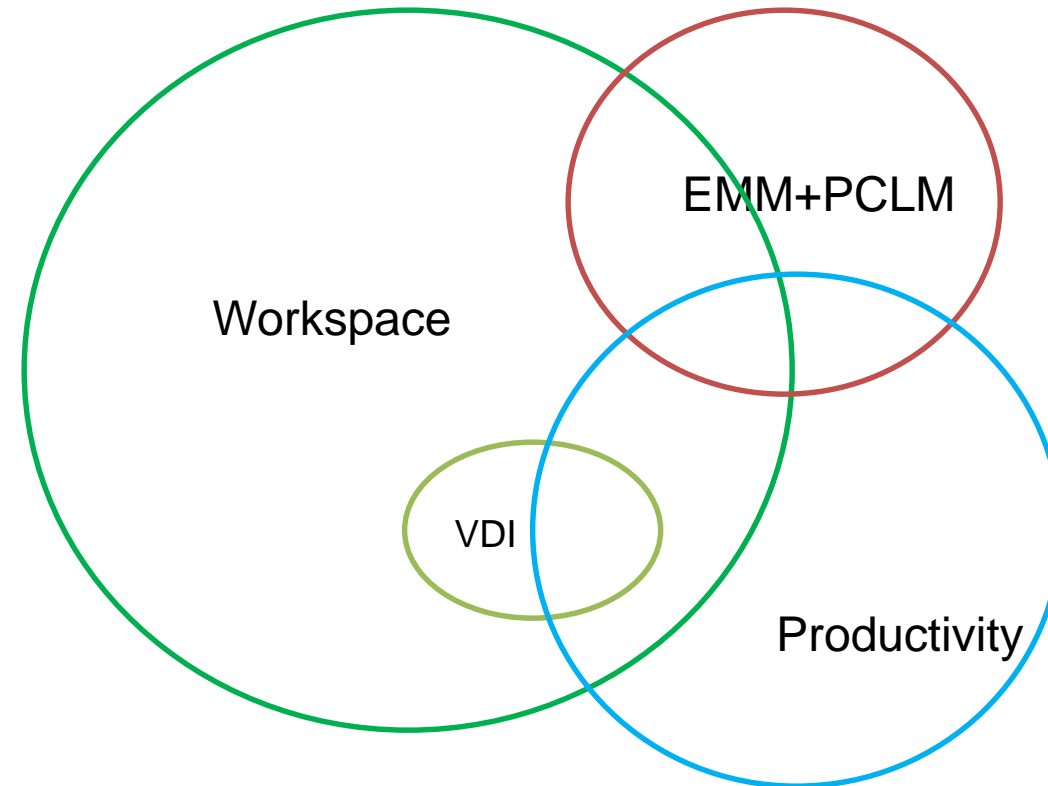


# The Road Ahead

EUC will be

- Multi-tenant
- Cloud Control
- Built for modern data center

**VDI will be build for cloud  
and hyper-convergence**



# Conclusion

Till now VDI has been low due to the high storage and management costs

- With HCI, that is set to change
- HCI allows easy scalability as compared to traditional 3-tier infrastructure
- Newer VDI solutions makes the VDI layer much thinner by leveraging the advanced features of modern Hypervisors and HCI

HCI support demanding VDI requirements by offering essential scalability, improved visibility and cost predictability



Thank You

*“It is not the strongest or the most intelligent who will survive but those who can best manage change.” – Charles Darwin*

*Welcome to possible*

Amit Motiwale  
Practice Head  
[amit.motiwale@mindtree.com](mailto:amit.motiwale@mindtree.com)